AUTOSTAT™: OUTPUT STATISTICAL ANALYSIS FOR AUTOMOD™ USERS

John S. Carson II

AutoSimulations
1827 Powers Ferry Road
Bldg 17, Suite 100
Atlanta, Georgia 30339, U.S.A.

ABSTRACT

AutoStat™ is an extension package for AutoMod™ and AutoSched™ models that provides complete support for simulation model experimentation and statistical analysis of outputs. Within a menu-driven, point and click environment, AutoStat provides assistance for setting up runs, automated execution of runs and consolidation of outputs across replications and scenarios. AutoStat automatically sets random number seeds to achieve statistically independent replications; it also sets factor levels (input parameters) to realize desired scenarios, without having to modify the underlying model.

AutoStat has a number of advanced features such as datafile factors, responses (or performance measures) read from custom user output files, and user-defined combination responses (linear combinations of other responses). AutoStat offers several statistical methods, including confidence intervals, a ranking and selection procedure, design of experiments, and warm-up determination. Outputs can be displayed and printed in tabular and graphical formats and exported to spreadsheet and other analysis software.

AutoStat saves simulation modelers considerable time and automates the effort of setting up and making runs, managing the input and output files, consolidating results and conducting analyses.

1 INTRODUCTION

The experimentation phase of simulation modeling and analysis is critical to the success of any simulation modeling project. AutoStat provides assistance for all parts of the experimentation phase, including:
- Setting up model runs
- File management for the model's input and output files
- Defining the desired scenarios to be run
- Automatic execution of desired scenarios, including control of random number streams, setting of factor levels (input parameters), and error tracking
- Consolidation of outputs across scenarios and across statistical replications
- Statistical calculations across replications, including sample means and standard deviations
- Statistical methods, including:
  Confidence intervals
  A ranking and selection procedure
  Design of experiments
  Warm-up determination
- Tabular and graphical display and printing of results
- Export of tabular outputs to spreadsheets and other analysis software

AutoStat works with both AutoMod and AutoSched models. AutoMod is a general purpose simulation package with 3-D animation that provides pre-programmed, material handling simulators for conveyors and vehicle systems and special support for manufacturing systems. AutoSched is a simulation-based, finite-capacity production scheduling tool.

AutoStat provides all the tools you need for setting up runs and conducting a proper statistical analysis, in an easy-to-use point and click, menu-driven environment. AutoStat provides warm-up determination for steady-state analyses, single and multiple-comparison confidence intervals for the estimation and comparison of mean system performance measures, the design of experiments to determine the significant factors, and a ranking and selection procedure ("select the best") for finding the "best" system. Statistical results can be displayed in both tabular and graphical formats. Responses, or system performance measures, that can be analyzed include all those in the standard AutoMod output report plus responses printed to custom output reports. In addition, you can define a response as a linear combination of other, previously defined responses, without modifying the underlying model.

In support of experimentation and analysis, AutoStat provides automated set-up and execution of model runs, file management and consolidation of outputs across runs. You define the factors or input parameters that you want to vary. Each set of factor values defines a different scenario or system alternative of interest. Possible factors include any standard AutoMod input parameter (such as number of vehicles or operators, conveyor speed, or vehicle speed), any user-defined variable's initial value, or any user input data file (such as a production schedule or daily orders).

After you identify the factors you want to vary, define the scenarios of interest, and specify the number of replications desired, AutoStat will automatically make all the runs for all desired scenarios and archive the outputs for later analysis and display of comparative results. For each run, AutoStat automatically sets the seeds for all random number streams, and sets user-specified values for factors, including data file factors, without modification to the underlying model.

In support of debugging and verification, AutoStat allows previous runs to be repeated interactively, so that suspicious outputs can be investigated and the model verified and errors corrected. When repeating a run for debugging purposes, AutoStat automatically duplicates all previous settings, including random number seeds and factor values. Models with errors can be corrected and previous runs re-made for verification purposes.

If your model has probabilistic or statistical components, such as random arrivals or random processing times, then you need AutoStat for conducting a statistical analysis to determine whether observed differences in a response between two scenarios or two system alternatives are real differences or are merely due to random fluctuations. If your model is deterministic, then AutoStat can still save you considerable time and effort in setting up and making runs across numerous scenarios.

After AutoStat makes all the runs you request, you select the runs to compare, choose the type of analysis or graphical comparison, and AutoStat will extract the relevant outputs, do the statistical analysis and display the comparative results automatically. AutoStat can compare alternatives on the basis of any standard AutoMod output or any user-computed output written to a custom output report. Responses can be added after runs have been completed and analyses conducted on these responses, without having to re-run the model.

AutoStat makes it relatively simple to setup and run a large number of comparative simulations. With a few selections with the mouse, you can get results comparing the model's performance measures across any number of alternative scenarios. Results can be displayed in graphical and tabular form. In fact, without using the statistical capabilities at all, AutoStat can save you a tremendous amount of time during the debugging and verification phases of modeling, and thus help to make your model more robust. In addition, when setting up and making a set of runs for comparing alternatives, AutoStat can save you a tremendous amount of time compared to using batch files and manually managing all of the input and output files.

The remainder of this paper is organized as follows: Section 2 covers the basic concepts for setting up runs in AutoStat, namely, responses, factors, configurations and sample types. Section 3 explains the menu system. Section 4 covers the statistical procedures available in AutoStat. Section 5 discusses the use of AutoStat during model debugging and verification, and Section 6 discusses it use during experimentation. Section 7 provides a summary.

2 BASIC CONCEPTS

To use AutoStat to make runs for the purpose of experimentation or model verification, you begin by defining responses, factors, configurations and sample types. First, you define the responses or measures of system performance; second, you define the factors, or input parameters, that will be varied to generate alternative scenarios or system designs to be compared. A set of factors is grouped into a configuration, or experimental framework, to define the desired scenarios to be simulated. For each factor in a configuration, you then define the desired value or range of values, and AutoStat automatically generates the requested runs. You also must specify the sample type, namely the warm-up period and runlength, plus the number of statistical replications. Then, AutoStat takes over, makes all requested runs, and archives the results for later analysis. This section discusses these basic concepts.

2.1 Responses

A response is a model output, usually a measure of system performance. For example, typical responses include machine or worker utilization, average and maximum queue size, and throughput. A response can be any output statistic captured by AutoMod or AutoSched and printed to any of the standard output reports.

AutoStat also allows user responses and combination responses. Any output statistic printed to a custom output report is called a user response. AutoStat can be "taught" to read a custom output report and "find" the desired output to be used as a response. To achieve this, the custom output report must contain a "snap separator", that is, a unique string that identifies the

beginning of the custom report for the warm-up period (if any) and the beginning of the report for the steady-state portion of the run. For models without a warm-up period, the snap separator can be any string at the top of the output file. The custom report also must contain one or two unique strings of characters, each usually a header or title, before the desired statistic, either on the same or a preceding line as the statistic. AutoStat is then told to look on the same line as, or a specified number of lines after, the unique character string, and to read a specified column from that line (counting columns starting at 0). You must also tell AutoStat whether the statistic is an average, a total (such as counting the occurrences of some event), a minimum or a maximum, so that AutoStat can handle any subsequent analyses properly.

Identification of AutoSched responses has been automated even further. Standard AutoSched output files are displayed in an edit table format, similar to a spreadsheet, with rows and columns. Clicking with the mouse on a desired element identifies the output in that row and column as a response.

Combination responses are new responses defined as linear combinations of other responses. This allows responses such as cost functions or averages to be computed by AutoStat without changing the model in any way.

For each run made, AutoStat saves the complete AutoMod report plus all custom reports requested to an archive in a compressed format (to minimize usage of hard disk space). Because of the archiving of all outputs, additional responses can be defined after a set of runs have been completed, without having to re-run the model, and additional analyses conducted.

2.2 Factors

A factor is a model input parameter that can be varied during experimentation. Usually, a set of factors set to particular values defines a scenario or alternative system design of interest. Examples of factors include the number of workers, product flow, machine speeds, production schedules and operating rules, order files, conveyor speeds, and vehicle performance specifications (speeds, accelerations).

In an AutoMod model, the following items can be factors:
- Capacity of a resource, counter, block or any other built-in entity.
- Speed or other performance parameter of a vehicle in an AGVS, ASRS, Bridge Crane, Power & Free or other vehicle system.
- Capacity of a vehicle (number of loads that can be transported).
- Speed of a conveyor section.

- Size of a load (important for conveyor models).
- Initial value of a variable.
- External input data file.

In addition to these factors, a special type of factor is available for AutoSched models. All AutoSched models are driven by a standard set of input tables defined through a spreadsheet-like user interface. Examples include Station, Routing and Calendar files. When defining an AutoSched factor, AutoStat displays the actual input table and you may select one or more cells as factors. This means that you desire to change the values in this group of cells during experimentation. In addition to cells, rows to be modified or added can be defined as AutoSched factors.

When any of these parameters is designated as a factor and assigned values, AutoStat will automatically override their value in the model (called the base value) without any more effort on your part than choosing the factor and typing the desired factor values.

Operating logic can be made a factor by using an input variable as a flag, for example to indicate which set of operating rules to use, and having the model select which rules to use based on the variable's value. In this way, alternative operating rules or logic can be compared against each other.

As mentioned, an external input data file can be defined as a factor. Examples of potential data file factors include production schedule files, product routing files, order files, machine processing times and downtime data. To use a datafile factor, first you create two or more alternative input files. Then by point and click you tell AutoStat that the input file is a variable factor and specify the alternative file names. For example, suppose a model of a distribution center reads a file of daily orders called "order.d". You obtain samples of order files for 5 different days over the past year, calling them "order.1", "order.2", ..., and "order.5". You tell AutoStat that "order.d" is a datafile factor and specify the names of the alternate order files. When such a file is used as a factor, AutoStat automatically copies the alternate files to the filename hard-coded in the model (saving the original and restoring it after all runs are completed). No changes to the model itself are required.

2.3 Configurations and Sample Types

Defining configurations and sample types is the next step in setting up experimental runs. AutoStat uses the concept of a "configuration" to designate a set of factors to be varied for a particular analysis. Other factors (if any) are given a constant value, either the value hard-coded in the model or any other desired value. A

configuration thus provides an experimental framework for a group of experiments.

Next, you define the sample type or run conditions. For example, for a particular model, it might be appropriate to have an 8 hour warm-up period and a 40 hour run. (For steady-state simulations with an initialization phase, AutoStat can assist in determining the appropriate warm-up period, as described in Section 4.1.) For a terminating (non-steady state) simulation, you simply define the desired runlength and a zero-length warm-up period.

2.4 Setups

To setup the sample runs for a configuration, you define the value or range of values for each variable factor. You can setup runs one at a time by specifying values for each variable factor in a configuration, or you can setup multiple runs at once by defining the set of values for each factor and have AutoStat generate all combinations of factor values. Finally, you define the desired number of statistical replications, with three being the minimum recommended number. Each statistical replication of a scenario uses the same factor values but uses a unique stream of random numbers to guarantee statistical independence and random sampling, as required by most statistical methods.

2.5 Making Runs

After defining factors and responses, configurations and sample types, and setting up the desired sample runs, you make the model runs by clicking on the Run menu and selecting "Do all runs". AutoStat will automatically make the requested number of independent replications of all the scenarios that you have set up.

For example, suppose that in a model of a production line, the following variables are the experimental factors, with factor values as given:

Number of Packing Machines: 3, 4, or 5
Number of Operators per Coloring Machine:
      1 operator per 1 machine
      1 operator per 2 machines
Production Schedule (a datafile factor):
      "SCHED1.DAT"

      "SCHED2.DAT"

AutoStat will automatically make the runs for each requested combination of factor values, and save the standard AutoMod output report and all custom reports that contain responses. In this example, for one replication per scenario and using all combinations of

factor values, each set of runs amounts to $3 \times 2 \times 2 = 12$ runs. With 3 replications, AutoStat would make 36 runs. Multiple replications are needed by most statistical procedures.

3 MENUS AND MECHANICS

With Version 8.0 and higher, AutoStat and other AutoSimulations' products are true Windows™ products for the PC, and are based on the Motif™ graphical user interface on UNIX platforms.

To set up and make runs, you use the Model, Responses, Factors, Configurations, Samples and Runs menus. The Model menu is for opening models and saving AutoStat default data. With these menus, you setup all AutoStat experiments and get many of the benefits of using AutoStat in terms of data and scenario management.

The other menus cover the statistical and display capabilities:

1.   Warm-up Determination:
        Setup warm-up runs and display Welch moving average plots of responses over time.
2.   Confidence Intervals:
        Single and comparison intervals, the "select the best" ranking and selection procedure.
3.   Design of Experiments:
        Setup DOE runs, and graph and print DOE results.
4.   Graphs:
        Plot of responses versus a factor
        Confidence interval plot of response for each configuration

4 STATISTICAL PROCEDURES IN AUTOSTAT

In this section, we describe the statistical capabilities of AutoStat. These are divided between Warm-up Determination in Section 4.1 and the analyses in Sections 4.2 - 4.5. The types of analyses offered include confidence intervals, a 2^k design of experiments, and a ranking and selection procedure called "Select the Best". (For more background on statistical methods in simulation, see Banks, Carson, and Nelson (1996) and Law and Kelton (1991).)

4.1 Warm-up Determination

If you are interested in the steady-state behavior of a system and not the warm-up or initial transient caused by starting the simulation in an empty and idle condition or some other state not representative of steady state, then you need to determine the length of the warm-up or initial transient phase. After determining a reasonable

length for the warm-up period, you will run the simulation for a reasonably long period after the warm-up phase, so that system performance statistics (responses) can be collected over a period that is close to steady state conditions. It is recommended that the steady-state phase be at least twice as long as the warm-up period, since the warm-up period is usually related to the (maximum) length of time it takes for parts to get through the system. Having a warm-up period that is too short, failing to load the system to a state representative of steady state, or failing to run the steady-state phase long enough, could result in performance statistics that are biased low due to the empty and idle initial state at time zero and that are not good estimates of the system's steady state behavior.

For warm-up determination, you should make a long run of one or more scenarios, breaking the run into a large number of short snaps. As a rule of thumb, we recommend 100 snaps. In AutoMod, a snap is a reporting period; standard output reports are written to a file at the end of each snap, statistical accumulators are reset, and the simulation continues. If you tell AutoStat to make 100 snaps for warm-up determination, you will have a plot based on 100 response values, one response value for each snap period.

You should verify that all parts of the system become loaded and process a reasonable number of loads during the specified runlength. To determine whether it is reasonable to assume that the model has reached a steady state, you should examine all key responses and how fast, or whether, they are converging to a steady state. In addition, you may want to define a number of special responses for detecting that all subsystems, or all entity types, have been successfully loaded. For example, you may want to define responses for waiting times at the end of any product routing, or throughput by product type, to be sure that the whole system is loaded or that low volume or slow moving product has reached a steady state.

There is no known reliable method for automatically determining, by a statistical or mathematical calculation, when (or whether) a model has reached steady state. The best methods are graphical and depend upon your judgment and your knowledge of the actual system.

To assist in determining the length of the warm-up period, AutoStat provides a moving average plot of any selected response over the snaps. To reduce random fluctuations or noise, the plot is averaged in two ways: across replications and within replications. The across-replication averaging is automatic. The within-replication average is the moving average plot, defined by a user-adjustable window size.

A moving average is the average over several adjacent snaps of the response value for those snaps. The number of adjacent snaps used to compute the moving average is called the window size. The purpose of the moving average is to smooth the plot in order to detect the underlying trend or low frequency fluctuations over time by dampening the random noise or high frequency fluctuations. Having a large number of short snaps makes the plots more meaningful, easier to interpret, and also allows for more flexibility in choosing a window size. For example, if you request 100 snaps and a window size of 1, the plot will consist of 100 response values. If you request a window size of 3, then snaps 1-3 are averaged, snaps 2-4 are averaged, ..., and snaps 98-100 are averaged, resulting in 98 values to be plotted.

To determine a reasonable warm-up period, you must use not only the moving average plots with a suitably chosen window size, but also your knowledge that all areas of the system have reached a reasonable loading. It should be kept in mind that some configurations may not have steady state, usually indicated by a response value that continues to rise in value the longer the model is simulated. The AutoStat moving average plot can assist in determining whether a steady state is reached at all.

After you have determined the warm-up period, you setup a sample type with the warm-up and steady-state runlengths desired, and AutoStat will automatically ignore the responses over the warm-up and use responses over the steady state for all analyses.

4.2 Confidence Intervals

A confidence interval for a single scenario is used to estimate long-run mean value of a response. It measures the effect on the estimate of mean response due to inherent statistical variation in the model. As more replications are made, the width of the confidence interval decreases, indicating greater certainty in the estimate in a statistical sense. (A short confidence interval does not, of course, say anything about a model's validity.)

A confidence interval for the difference in response between two scenarios is used to make a judgment regarding whether, and how much of, the observed difference is due to systematic differences in the two scenarios versus being due to mere random variation or noise.

To compute a confidence interval for a single scenario, AutoStat computes the overall average response over all replications, computes the sample standard deviation of the responses, and the standard error of the overall average response. Then it computes the confidence interval at the desired level of confidence (usually, 95%). The confidence interval is an interval around the overall average response value that estimates uncertainty in the estimate of mean response; it provides

an estimate of the amount of variation in the overall sample average response due to statistical variation in the inputs for the particular runlength used.

A confidence interval should be used only when the response of interest is an average; it should not be used for maximums or minimums. This is because of the statistical theory (normality theory and the Central Limit Theorem) on which the computations are based. In fact, AutoStat will not compute confidence intervals for statistics of type minimum or maximum.

Confidence intervals are perhaps more useful when comparing the response of two (or more) scenarios. Suppose two alternative system designs (A and B) of a widget production facility were simulated for 4 replications and average widget time in system was compared. Suppose, further, that AutoStat reports a 95% confidence interval on the difference in time in system between systems A and B as:

$$3.2 \pm 3.8 \text{ hours, or}$$
$$-0.8 \text{ to } 7.0 \text{ hours.}$$

Since the interval contains zero, and thus overlaps both negative and positive ranges, no conclusion can be drawn as to whether system A or B is better, at least on a statistical basis. However, if a 4 hour improvement for widget time in system was of no practical importance, it can be concluded that neither system is better than the other. On the other hand, if a 2 hour difference was of importance, then more replications would be needed to reduce the width of the confidence interval so that a more precise estimate could be obtained. With a sufficient number of replications, the confidence interval should lie entirely to the left, or entirely to the right, of the value 2, in which case strong conclusions can be drawn.

If a tighter estimate of average difference were desired, how many more replications would be needed? In general, to decrease the confidence interval width by half, it is necessary to make 4 times the original number of runs. For example, for the widget example with 4 runs, it would take 12 more runs for a total of 16 to get a confidence interval of roughly half the original width (that is, of roughly $\pm 3.8/2 = 1.9$ hours in width).

4.3   Design of Experiments

One method for determining which factors have a statistically significant effect upon any response is to apply AutoStat's design of experiments. Statistical significance for a factor means that with high confidence the observed variation in mean response as the factor is varied from a low to a high setting is more than can be accounted for by mere random variation; that is, the observed variation of the response is larger than what would be expected due to the random variation alone.

Thus, the factor is judged to have a significant effect on the response value. With this procedure, you can quickly eliminate from further experimentation many factors that have little or no effect on the important system performance measures, and use future computer and analysis time more profitably by concentrating on the factors most likely to have a large effect on system response.

AutoStat offers the so-called 2^k design; each factor is allowed two values, a "low" and a "high" setting. AutoStat offers the full factorial design, requiring all combinations of factor levels to be simulated. For example, for k=5 factors, a total of $2^k = 32$ runs are needed for each replicate of a full factorial design. AutoStat also offers several lower resolution designs, allowing a trade-off between number of runs and the possible inferences that can be drawn reliably.

4.4   Select the Best

AutoStat also offers a "Select the Best" ranking and selection procedure, one of the subset selection procedures presented in Law and Kelton (1991).

The goal is to select the best system, defined as the one with the minimum or maximum value of some specified response, from among k systems. For example, the best system might be the one with maximum throughput or minimum average time in system per part produced. The procedure can select either the single best system, or a subset containing the best system. The subset size is designated by m. For $m = 1$, the single best system is desired. For $m > 1$, the procedure will select a subset of size m containing the best system.

Besides specifying k and m, a user specifies the minimum probability of correct selection, P^*, and an indifference amount, d^*. Typically, $P^* = 0.95$, and the goal is $P(CS) = \text{Prob(Correct Selection)} \geq P^*$.

The choice of indifference amount, d^*, depends on engineering or practical considerations. If the response of a system is very close to the response of the best system, then we do not care which of the two systems is chosen as best. More precisely, if the two system responses are within d^*, then we are "indifferent" to which one is chosen as best. With this setup, we are guaranteed that $P(CS) > \geq P^*$, where correct selection is defined as the best or any system within the indifference amount of the best.

The ranking and selection procedure in AutoStat is a two-stage procedure. First, an initial sample is taken of at least 5 runs of each configuration being compared. This initial sample is used to provide preliminary sample means and sample variances, and to determine the number of additional samples needed to select the best subset with the specified probability of correct selection.

Note that *P** and *d** can be selected after the first-stage of sampling; the advantage of this is that AutoStat computes and displays the number of additional samples needed for a specified *m*, *P** and *d**. In general, the final sample size, N_i, required for each configuration *i* gets larger as subset size *m* or indifference amount *d** decreases, or as probability of correct selection *P** increases. It is recommended that *P** be set at 95% and *m* and/or *d** be made larger if the required sample size is too large for the available computer time.

Theoretically, the "Select the Best" procedure could be used in a factorial experiment to automatically find the combination of factor values that results in an optimum response. But in practice, with even a moderate number of factors and factor values, the procedure may require such a large number of replications as to exceed available computer time. Therefore, it is recommended that as many factors as possible be eliminated (and set at some appropriate constant value) by using either the Design of Experiments or multiple comparisons, or perhaps Select the Best with a moderately large subset size. Then Select the Best can be used with a small number of the most significant factors to determine the optimum combination of factor values.

4.5   Graphical Displays

AutoStat provides graphical displays of one or more responses versus a factor, and plots of confidence intervals for a selected set of configurations. In addition, the results of the Design of Experiments can be displayed graphically, namely, the confidence intervals for the main and interaction effects.

5   USING AUTOSTAT DURING MODEL TESTING AND VERIFICATION

Even before using its statistical capabilities, AutoStat can be of great assistance during model testing and verification. In brief, AutoStat provides a quick way to setup and run a large number of tests overnight or over a weekend, and a quick and convenient way to view the resulting outputs for face validity.

Key parameters can be varied easily by selecting them as factors. In fact, in a brief 10 to 15 minutes for setting up runs, almost any number of test runs can be setup and started. For verification purposes, you should make at least 3 statistical replications and run as wide a range of scenarios as possible, given the available computer time.

When you return to view the results, most likely AutoStat will have finished all the runs. If not, simply click on the Cancel button to abort the remaining runs. Completed runs will be saved and canceled runs can be completed at a later time if desired. For each run made, the usual AutoMod message window is on the screen; it will tell you whether the run completed the requested runlength or stopped with a modeling error. Of course, your model may also print messages to the message window, perhaps reporting some logical error detected during the run.

Using the Display Samples menu item on the Sample menu, you can display a table of all responses over all runs. A quick visual scan of this table will usually indicate whether the responses are consistent within a scenario. For example, looking at throughput for the whole system and key subsystems will tell you whether the model has a "lock-up" condition indicating a modeling error. The purpose here is not a detailed analysis, but rather to incorporate model verification into model development in an ongoing fashion and to provide more robust model testing by simulating over a wide range of factor settings.

If a run ends with an error or some unexpected output, you can use AutoStat to easily repeat that particular run in normal interactive mode to find the cause of the problem. You may use the debugger, watch the animation, or use any other debugging technique. Choose Repeat Run on the Runs menu, select the run you want to repeat, and AutoStat will automatically set the factor values and the random number seeds, so that you can repeat the previously made run. After finding suspected errors and modifying the model to fix them, you can again use AutoStat to repeat the previous run in order to verify the model's correctness.

Usually, after modeling errors have been fixed and the model has been satisfactorily verified, you should delete all runs made for verification and debugging purposes. After all, the model has probably changed during this period (to fix errors) and experimentation should be done with the final verified model.

The main purpose of using AutoStat in these early phases of a project before experimentation begins is as a quick test that a model runs to completion, and that the outputs (responses) are reasonable, over a wide range of possible inputs. This provides much more robust testing over a wider range of conditions than can be achieved by making one run at a time manually.

6   AUTOSTAT AND EXPERIMENTATION

AutoStat was designed to provide a set of basic statistical procedures to assist during the experimentation phase of a project. The data management and scenario management capabilities allow you to have more time to concentrate on the analysis itself instead of the tedious details of setting up runs and keeping track of all the output files.

The main purpose of the statistical procedures in AutoStat is to allow you to conclude with reasonable confidence that observed differences in response between two (or more) system designs or scenarios are due to a true systematic difference and not merely due to random noise. When a model has one or more input variables defined as a statistical distribution, say, for example, a time to failure that is exponentially distributed with mean of 6 hours, then random noise in the responses is unavoidable. Each independent replication uses a different stream of random numbers, the result being observed variation in the output responses. If there is a difference in the response of two system designs (or scenarios), is it a true systematic difference or merely noise? The statistical procedures built into AutoStat can assist in making a sound judgment.

Since AutoStat saves the standard AutoMod output report and all designated custom reports (in compressed form to minimize disk space usage), you can add a new response after making the simulation runs. This comes in especially handy when your boss or customer requests a new or different measure of system performance; it avoids the necessity of repeating runs previously made.

7 SUMMARY

As with any analytic tool, it's up to you to interpret the simulation results properly and determine the practical significance of the factors on the responses. To leave more time for analysis, AutoStat can save you a tremendous amount of time by making it easier to setup and make any desired number of test runs and experimental runs. AutoStat will automatically save all the desired outputs and present them to you with a mouse click, in both tabular and graphical formats. For statistical analysis, AutoStat finds the desired outputs across any number of scenarios, does all the required statistical calculations, and, with a few clicks of the mouse, presents the statistical results in tabular and graphical formats.

ACKNOWLEDGMENTS

This paper is an updated and modified version of a paper of the same name by the same author that appeared in the *Proceedings of the 1996 Winter Simulation Conference* (John M. Charnes, Douglas M. Morrice, Daniel T. Brunner, James J. Swain, eds.), pages 492-499.

REFERENCES

AutoStat User's Guide, *AutoMod User's Manual, Volume II*. 1996. Bountiful, UT: AutoSimulations.

Banks, J., J.S. Carson, and B.L. Nelson. 1996. *Discrete-Event System Simulation*. Second Edition. Upper Saddle River, NJ: Prentice-Hall.

Carson, J.S. 1996. AutoStat: Output Statistical Analysis for AutoMod Users. In *Proceedings of the 1996 Winter Simulation Conference,* ed. John M. Charnes, Douglas M. Morrice, Daniel T. Brunner, James J. Swain, 492-499. San Diego, CA: The Society for Computer Simulation.

Law, A.M. and W.D. Kelton. 1991. *Simulation Modeling and Analysis.* Second Edition. New York: McGraw-Hill.

AUTHOR BIOGRAPHY

JOHN S. CARSON is the Regional Consulting Manager for the Atlanta office of AutoSimulations. With ASI since 1994, he has over 20 years experience in simulation in a wide range of application areas, including manufacturing, distribution, warehousing and material handling, transportation and rapid transit systems, port operations and shipping, and medical/health care systems. His current interests center on the simulation of transportation systems, train systems, bulk and liquid processing, and software for simulation output analysis. He has been an independent simulation consultant, and has taught at Georgia Tech, the University of Florida, and the University of Wisconsin-Madison. He is the co-author of two textbooks, including the widely-used Discrete-Event System Simulation, Second Edition (1996). He holds a Ph.D. in Industrial Engineering and Operations Research from the University of Wisconsin-Madison, and is a member of IIE and INFORMS.